



# **DIME DYNAMIC DOCUMENTATION TRAINING**

## **Exercise 2**

Luiza Andrade & Mrijan Rimal

June 8, 2017

---

## Introduction

Exercise 1 introduced you to the basics of how to import tables and figures to a  $\text{\LaTeX}$  document. This exercise will introduce you to some intermediate topics commonly used to make your document look even more professional.

We will also show how  $\text{\LaTeX}$  can be used to create a dynamic document that updates automatically once your output from, for example, Stata or R is updated without any error prone manual copy-and-pasting.

## Part 1. Exporting tables

In this section, you have the option of using the template do file provided in [Dynamic-Documentation/Exercises/Stata Export Exercise/Export tables and images.do](#) to export graphs and figures OR using your own data and generating tables and graphs that can be imported into  $\text{\LaTeX}$ .

If you're using the do file provided, change the folder paths in the do file to your directory structure and run the do-file. This will export two tables and two graphs to the Raw folder that will be used in the exercises that follow.

If you would like to work with your own data, please follow the following steps.

- Make sure that you export **two tables** and **two graphs** using your own data. For help, you can look at the folder called “Exercise Stata - How to export tables and graph from Stata to LaTeX”.
- Make sure that all the exporting is done using Stata do-files, or R scripts so you can easily export them again by just running the code. Towards the end of this exercise we will ask you to make changes to your do-file or R script so if you are using a file from an actual project you may want to make a copy of your do-file or R script that you can use in this exercise.
- Import all the tables and figures you have created in step 1 into your  $\text{\LaTeX}$  document. Sample code to import tables and figures can be found in the **DIME Templates** sub-folder. Make sure to use the code in **Template 1 - Importing tables.tex** file to import figures(not tables), and the code in **Template 2 - Importing figures.tex** file template to import graphs(not tables). It will not work if it is the other way around.

- 
- Make sure you give a caption to your tables.

## Part 2. Intermediate L<sup>A</sup>T<sub>E</sub>X Exercises

### Part 2.1 Adding Sections to your document

L<sup>A</sup>T<sub>E</sub>X automatically formats the document and different section headers and subheaders according to pre-defined formats. It also allows you to automatically create a beautiful **Table of Contents** based on what has been defined as sections and subsections.

Sections can be created using `\section{title}` command. L<sup>A</sup>T<sub>E</sub>X automatically numbers all the sections in the order you put them. Since, you manually don't specify chapter and section numbers, you can cut and paste subsections from the end to the beginning and the numbering will update automatically!

Similarly, subsections can be created using `\subsection{title}` and sub-sub-sections can be created using `\subsubsection{title}`. The subsections will be numbered on the format 1.1 and the subsubsection will be numbered on the format 1.1.1. An example is shown in Annex 1.

*Note: Sections and subsections can be created using `\section*{title}` if you do not want your sections numbered in the document. However, these sections and subsections will not be shown in the table of contents.*

---

## Part 2.2. Adding Table of Contents

After we have set up sections, sub-sections and sub-sub-sections you can easily add a table of contents to your L<sup>A</sup>T<sub>E</sub>X document by using `\tableofcontents` in your document. You can create this any where you want, but typically this is created directly after `\begin{document}` or after `\maketitle` if you have a title. An example is shown below:

```
\documentclass[12pts]{article}

\title{My Awesome Document}
\author{John Doe}

\begin{document}
\maketitle
\tableofcontents
\newpage
```

The above L<sup>A</sup>T<sub>E</sub>X code is used to generate table of contents below.

My Awesome Document		
John Doe		
May 31, 2017		
 <b>Contents</b>		
<b>1</b>	<b>This is a test section</b>	<b>2</b>
<b>2</b>	<b>This is the second test section</b>	<b>2</b>
2.1	This is a test subsection . . . . .	2
2.1.1	This is a test subsubsection . . . . .	2

Figure 1: Example table of contents generated from our document in Annex 1.

---

## Part 2.3. Referring to your tables, and figures in a document

Just like the numbering of sections and sub-sections, L<sup>A</sup>T<sub>E</sub>X uses a dynamic referencing system for tables and figures etc. In the text in your document where you describe your tables and figures you are likely to want to refer to them on a format similar to "*As you can see in figure 2...*". Since the numbering of tables and figures is updated automatically, you need a way for your references in your text to be updated automatically as well.

In L<sup>A</sup>T<sub>E</sub>X that is solved by giving tables and graphs a unique name using the `\label{}` command. In the text where you want to reference a figure or a table, you reference the name used in the label and L<sup>A</sup>T<sub>E</sub>X will update the numbering for you as your documents grows and changes.

The labels are categorized into the type of item you are referencing, so you label tables and figures slightly differently. We will show how it is done in the following sub-sections.

### Part 2.3.1 Referring to Figures in a document

To refer to figures, we will use the `\label{fig:figurename}` command. Inside the brackets in the `\label{}` command you see two parts separated by a colon. The first part `fig` indicates that this is a figure that we are assigning a label to. The second part `figurename`, you should replace with the unique name you want to use to refer to this figure.

We must add the `\label{fig:Figurename}` line inside the `\begin{figure}` and `\end{figure}` in our codes. See below:

```
\begin{figure}[H]
\includegraphics[width=\textwidth]{../Raw/iegraph.png}
\caption{Add figure title here}
\label{fig:iegraph}
\end{figure}
```

Now to refer to the figure anywhere in the document, you can type `\ref{fig:iegraph}` and it will automatically refer to the figure using the correct number in the document. In our example, if the `iegraph` figure is the fourth one in the document, *As you can see in figure \ref{fig:iegraph}* will appear as *As you can see in figure 4*, in the document.

---

### Part 2.3.2 Referring to Tables in a document

Similarly for tables, we can also refer to them anywhere in the document using the same method. To refer to tables, we will use `\label{tab:tablename}` command inside our `\begin{table}` and `\end{table}` like the example below.

First, we need to add a label to the table that has been imported to the document by adding `\label{tab:tablename}` inside the `begin` and `end` table in our code.

```
\begin{table}[H]
\caption{Add a title to this table}
\input{../Raw/sample_sizes.tex}
\label{tab:samplesize}
\end{table}
```

Now we can refer to the above table in our document by typing `\ref{tab:samplesize}` which will automatically refer to the table using the correct number in the text in your document.

## Part 2.4 Text Formatting

Now that you have created sections in your document, and added text, you might want to change some of the formatting of your text. In  $\text{\LaTeX}$  you are formatting your text using packages and code. We will here include a few topics.

### Part 2.4.1 Bold, italic and underlined

### Part 2.4.2 Color

### Part 2.4.3 Line Spacing

It is common that different publication standards require different line spacing. This can be achieved using the `setspace` package.

The use this package you need to start by importing it. To do import the package you add `\usepackage{setspace}` next to where a lot of other packages are imported using the same command `\usepackage{}`. Next

---

you simply to tell L<sup>A</sup>T<sub>E</sub>X which linespacing you want and you need to do this before `begin{document}`. `\singlespacing` makes L<sup>A</sup>T<sub>E</sub>X use single line space throughout the document. Similarly, `\doublespacing` and `\onehalfspacing` set the document to double line space and one half line space respectively. See example below:

```
\usepackage{setspace}
\doublespacing
\begin{document}
\maketitle
\section{Introduction}
This is the introduction paragraph
```

The `setspace` package can also be used to specify more particular linespacing i.e. 1.7 lines, 2.4 lines etc using the `setstretch` feature but that will be covered in the advanced exercise.

## Part 2.5. Rotating a table landscape

Sometimes the tables are very wide and need to be in landscape format. This can be adjusted using the `adjustbox` package which we have been using for importing tables. To rotate the table to landscape, we can use the `adjustbox` feature with the `angle = 90` option as shown in blue in the code below.

```
\begin{table}[H]
\caption{Add table title}
\begin{adjustbox}{angle = 90}
\input{sample_sizes.tex}
\end{adjustbox}
\label{tab:samplesize}
\end{table}
```

This will rotate the table to the specified degree in the final document.

---

## Part 3. Making a Dynamic Document

Here, we will produce a dynamic document. Please only do this if you have completed all tasks up to Part 2.5 of this exercise document and generated a document. This part will show you how easily L<sup>A</sup>T<sub>E</sub>X is updating your formatted documents without having to make any manual changes or manually copy and paste any files, code or results.

1. Make sure that you have generated a pdf document by compiling the your L<sup>A</sup>T<sub>E</sub>X code. Now copy the pdf created in a separate location from where L<sup>A</sup>T<sub>E</sub>X is exporting it. You will only need this second copy of your pdf document temporarily, so you can copy it to your desktop for example.
2. Make a significant change to your dataset by for example dropping a large subset of your data set. You want to make such a large change to your dataset that when you re-run your do-file or your R script, your outputs should look very different.

If you are using the Stata do-file template provided in the Dynamic Documentation Folder, then only keep observations with population growth more than 0. This can be done by using the Stata codes `drop if popgrowth < 0`.

3. Rerun the do-file or R script you used to generate the tables and files you imported to L<sup>A</sup>T<sub>E</sub>X in this exercise. Make sure to not change any file paths to where your tables and graphs were exported. You want your own tables and graphs to be replaced and overwritten.
4. Now, go back to the L<sup>A</sup>T<sub>E</sub>X code where you have imported the tables and graphs for this exercise and generate the pdf document again. Press *Build and Compile* under the *Tools*.
5. Now if you compare the pdf file you have just generated with you new data with the one you saved in a different folder in the beginning of this section, you will find that the tables have updated automatically without you having to do any manual edits.

In this part you updated your tables and figures in the final document without having to copy any results or files manually. Such manual updates are by far the most common reason for errors while generating and sharing research results. This is the most important reason why we at DIME are promoting L<sup>A</sup>T<sub>E</sub>X as manual errors causes us to lose a lot of time and in the worst case leads to mistakes in results that are published.



---

## Part 4. Challenge: Using a do-file to edit a .tex file after exporting it

*Note: This is an advanced exercise and you don't have to feel compelled to finish it. This is also best done after finishing all the other exercises in the GitHub repository.*

During this part of the exercise, you will learn how to use commands in Stata to format your tables. The reason for doing this is that we want to make as little edits in L<sup>A</sup>T<sub>E</sub>X as possible, so that when we update the tables like we did in Part 3, everything in the document file updates automatically. So, we would want to tweak or format our tables automatically using Stata, so that we do as little work in L<sup>A</sup>T<sub>E</sub>X as possible.

While tables exported from Stata to L<sup>A</sup>T<sub>E</sub>X are generally very nice, sometimes they need to be tweaked a little to make them look nicer. So, in this exercise, we'll use the `filefilter` command in Stata to make small changes to the files exported by Stata. `filefilter` is one such command in Stata which used to edit output files generated by Stata.

This exercise requires more familiarity with L<sup>A</sup>T<sub>E</sub>X than previous exercises. Don't worry if you can't complete it.

**Task 1:** Run the initial code for exercise 6 in the "Dynamic-Documentation/Exercises/Stata Export Exercise/Export tables and images.do" do-file. This will create a table with sample sizes for control and treatment groups across regions and in the whole sample. Add this table to the .tex file you created in the Exercise 2, Part 1. How does that look?

**Task 2:** Open the Dynamic-Documentation/Exercises/Exercise 1/Output/Raw/sample\_sizes.tex file created by Stata. Can you identify the source of the extra spacing?

**Task 3:** Use the `filefilter` command in Stata to filter out the lines or characters in the fragmented file that create the extra spacing. Import the new .tex file and check how it looks.

**Task 4:** Repeat task 3 if necessary.

---

## Annex 1 : Creating sections

### Commands typed in LaTeX document.

```
\section{This is a test section}
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque
laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi
architecto beatae vitae dicta sunt explicabo.

\section{This is the second test section}

\subsection{This is a test subsection}
Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci
velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam
aliquam quaerat voluptatem.

\subsubsection{This is a test subsubsection}
Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit
laboriosam, nisi ut aliquid ex ea commodi consequatur?
\end{document}
```

### Output

#### 1 This is a test section

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

#### 2 This is the second test section

##### 2.1 This is a test subsection

Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem.

##### 2.1.1 This is a test subsubsection

Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur?

Figure 2: Adding sections and subsections in a LaTeX document